ELSEVIER
Theoretical Computer Science 141 (1995) 253–268
Theoretical Computer Science

Multi-pattern languages☆

Lila Kari[a], Alexandru Mateescu[a], Gheorghe Păun[b], Arto Salomaa[a,*]

[a] *Mathematics Department, Academy of Finland and University of Turku, SF-20500 Turku, Finland*
[b] *Mathematics Institute of the Romanian Academy of Sciences, Str. Academiei 14, 70109 Bucuresti, Romania*

Received April 1993; revised January 1994
Communicated by G. Rozenberg

Abstract

We investigate languages consisting of words following one of the given finitely many patterns. The issues concerning such multi-pattern languages are relevant in inductive inference, theory of learning and term rewriting. We obtain results about decidability, characterization, hierarchies and special classes of multi-pattern languages. Some open problems are also presented.

1. Introduction

A natural way of describing a given sample of words is to exhibit a common *pattern* or *patterns* for the words. Such an approach is especially appropriate if the sample set is growing, for instance, through some learning process. Finding patterns for sample sets is a typical problem of inductive inference [8]. Languages defined by patterns are also closely related to word rewriting systems with variables [10].

Although the idea of patterns goes back to the seminal work of Thue [13] and was afterwards studied for instance in [3], pattern languages in the sense investigated in this paper were introduced by Angluin [2]. One starts with two disjoint alphabets, the alphabet Σ of *terminals* and the alphabet V of *variables*. A *pattern* α is a word over the union $\Sigma \cup V$. Thus, for $\Sigma = \{0, 1\}$ and $V = \{x, y, z\}$, $\alpha = 0x11xy$ is a pattern. A pattern defines a *language* consisting of words "following the pattern α". This means words obtained from α by uniformly substituting arbitrary terminal words for the variables. According to [2], the terminal words must be nonempty. We refer to this as the *nonerasing* or *NE*-case, α is then called also an *NE-pattern*. An essentially different theory results in the *erasing* or *E*-case [8, 9]. For instance, 01111 is in the language

☆ Research supported by the Academy of Finland, project 11281, and the Alexander von Humboldt Foundation.
* Corresponding author. Email: asalomaa@sara.utu.fi.

SSDI 0304-3975(94)00087-Y

defined by the E-pattern $\alpha = 0x11xy$ but not in the language defined by the NE-pattern α.

A natural way to generalize such *pattern languages* is to start with an arbitrary finite number of patterns instead of just a single one. In this paper we will investigate such *multi-pattern languages*. Indeed, in many cases no reasonable description of a sample set can be obtained using one pattern only. For instance, such a case results when the sample consists of lots of words with two different prefixes like 0001 and 1100. Then two patterns describe the sample much more appropriately than one.

A brief description of the contents of the paper follows. The basic definitions, as well as some initial results, are given in Section 2. Section 3 contains comparisons between multi-pattern languages and some other language families, namely, languages of simple matrix grammars [7] and languages of cooperating distributed grammar systems [4]. Such comparisons give results about the generative capacity of multi-patterns, as well as make it possible to transfer results concerning other languages to concern multi-pattern languages. Section 4 establishes an important undecidability result: it is undecidable whether or not a given context-free language is multi-pattern. The decidability status of the reverse problem (whether or not a given multi-pattern language is context-free) is open.

Section 5 deals with the hierarchy of language families obtained by increasing the number of patterns, and Section 6 closure properties of the family of multi-pattern languages. An important subclass, languages generated by repetition-free patterns, is investigated in Section 7. The concluding Section 8 contains some remarks about the ambiguity of pattern and multi-pattern languages.

This paper is largely self-contained. The reader is referred to [2, 6, 8–10] for more background and motivations, and to [12] for all unexplained notions in language theory.

2. Basic notions and preliminary results

Let Σ be an alphabet (of *terminals*) and let V be an alphabet (of *variables*) such that $\Sigma \cap V = \emptyset$. The set of words over $\Sigma \cup V$ is denoted by $(\Sigma \cup V)^*$ and the empty word is denoted by λ. A *pattern* α is word over $\Sigma \cup V$, i.e. $\alpha \in (\Sigma \cup V)^*$. Let $H_{\Sigma,V}$ be the set of morphisms $h, h : (\Sigma \cup V)^* \to (\Sigma \cup V)^*$.

We view patterns α as E-patterns (E from "erasing") and NE-patterns ("nonerasing"). The *language generated* by the E-pattern $\alpha \in (\Sigma \cup V)^*$ is defined as

$$L_{E,\Sigma} = \{w \in \Sigma^* \mid w = h(\alpha) \text{ for some } h \in H_{\Sigma,V} \text{ such that } h(a) = a \text{ for each } a \in \Sigma\}.$$

The language generated by the NE-pattern $\alpha, \alpha \in (\Sigma \cup V)^*$ is

$$L_{NE,\Sigma} = \{w \in \Sigma^* \mid w = h(\alpha) \text{ for some } \lambda\text{-free } h \in H_{\Sigma,V} \text{ such that } h(a) = a \text{ for each } a \in \Sigma\}.$$

If Σ is understood, we use also the notations $L_E(\alpha)$ and $L_{NE}(\alpha)$.

A *multi-pattern* π is a finite set of patterns, $\pi = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, $\alpha_i \in (\Sigma \cup V)^*$, $i = 1, \ldots, n$.

The language generated by an E-multi-pattern $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, $\alpha_i \in (\Sigma \cup V)^*$, $i = 1, \ldots, n$, is

$$L_{E,\Sigma}(\alpha_1, \ldots, \alpha_n) = \bigcup_{i=1}^{n} L_{E,\Sigma}(\alpha_i).$$

The language generated by an NE-multi-pattern $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, $\alpha_i \in (\Sigma \cup V)^*$, $i = 1, \ldots, n$, is

$$L_{NE,\Sigma}(\alpha_1, \ldots, \alpha_n) = \bigcup_{i=1}^{n} L_{NE,\Sigma}(\alpha_i).$$

We introduce the family of erasing multi-pattern languages of degree n as

$$MPLE(n) = \{L \mid L = L_{E,\Sigma}(\alpha_1, \ldots, \alpha_n) \text{ for some multi-pattern } \{\alpha_1, \ldots, \alpha_n\}\}$$

and the family of erasing multi-pattern languages as

$$MPLE = \bigcup_{n \geq 0} MPLE(n).$$

Analogously, the family of nonerasing multi-pattern languages of degree n is defined as

$$MPLNE(n) = \{L \mid L_{NE,\Sigma}(\alpha_1, \ldots, \alpha_n) \text{ for some multi-pattern } \{\alpha_1, \ldots, \alpha_n\}\}$$

and the family of nonerasing multi-pattern languages as

$$MPLNE = \bigcup_{n \geq 0} MPLNE(n).$$

We write also $L_{E,\Sigma}(\pi)$, $L_{NE,\Sigma}(\pi)$ for $\pi = \{\alpha_1, \ldots, \alpha_n\}$, $\alpha_i \in (\Sigma \cup V)^*$, $i = 1, \ldots, n$.

Lemma 1 (Jiang et al. [9]). *Let V be a set of variables, Σ be a terminal alphabet and $\Omega \subseteq \Sigma$. Consider a pattern $\alpha \in (\Sigma \cup V)^*$. Then there exist effectively $m \geq 1$ and patterns $\alpha_1, \ldots, \alpha_m \in (\Sigma \cup V)^*$ such that*

$$L_{E,\Sigma}(\alpha) = \bigcup_{i=1}^{m} L_{NE,\Sigma}(\alpha_i).$$

Consequently,

$$MPLE = MPLNE.$$

When the model E/NE is not relevant we write $L_\Sigma(\pi)$ instead of $L_{E,\Sigma}(\pi)$ or $L_{NE,\Sigma}(\pi)$. We write also briefly $MPL = MPLE$ $(= MPLNE)$.

There are essential differences between languages generated by E-patterns and NE-patterns, [8, 9]. For instance, while the *equivalence problem* is trivially decidable for NE-patterns (that is, the problem of whether two given NE-patterns generate the same language), its decidability status is open for E-patterns. Lemma 1 shows that, as far as the generated language families are concerned, there is no difference between E- and NE-multi-patterns.

Clearly, $L_\Sigma(\alpha) \subseteq L_\Sigma(\beta)$ iff $L_\Sigma(\beta) = L_\Sigma(\alpha, \beta)$. (This holds both for E- and NE-patterns.) Since the inclusion is undecidable [9] for pattern languages (both E and NE) and membership is NP-complete [2, 8] we obtain the following result.

Theorem 1. *The equivalence and the inclusion problems are undecidable for the family* $MPLE = MPLNE$. *The membership problem is* NP-*complete for languages in this family.*

One may consider *terminal-free* patterns, that is, words over the alphabet of variables. As regards single patterns, the inclusion problem is decidable in the E-case but open in the NE-case. As regards multi-patterns, the decidability of both equivalence and inclusion problems is open.

Instead of allowing arbitrary (uniform) substitutions for the variables, one may restrict the substitutions in various ways. A *generative* approach was taken in [6]. Initially one has a finite set of words that can be used in the substitutions. Whenever new words have resulted from the patterns, they become available for forthcoming substitutions.

Another possibility is to associate to each variable x a language $K(x)$, see also [1]; only words from $K(x)$ can be substituted for x. In the definitions above, $K(x) = \Sigma^*$ for E-patterns, and $K(x) = \Sigma^+$ for NE-patterns.

If $K(x)$ is regular for every variable x, we speak of *multi-pattern languages with regular substitutions*. Their family is denoted by MPL_{REG}. Clearly, we have the strict inclusion.

$$MPLE \subset MPL_{REG}.$$

3. Simulations of multi-patterns mechanisms

We now show that the family $MPLE$, in fact the family $MPLE_{REG}$, is contained in some other language families such as the well-known family $ETOL$ [11]. This gives an idea of the generative capacity of the mechanism of multi-patterns, as well as the possibility of applying to multi-pattern languages results concerning some other languages.

We begin with some further definitions.

Definition. A *cooperating distributed grammar system* (shortly a CD grammar system) is an $(n + 2)$ tuple,

$$\Gamma = (T, G_1, G_2, \ldots, G_n, S),$$

where (i) for $1 \leqslant i \leqslant n$, each $G_i = (N_i, T_i, P_i)$ is a context-free grammar with the set N_i of nonterminals, the set T_i of terminals, the set P_i of context-free rules, and without an axiom.
(ii) $T \subseteq \bigcup_{i=1}^{n} T_i$,
(iii) $S \in \bigcup_{i=1}^{n} N_i$.

The grammars G_i, $1 \leqslant i \leqslant n$, are called the components of Γ. Further we set $V_i = N_i \cup T_i$ and

$$V_\Gamma = \bigcup_{i=1}^{n} V_i.$$

Definition. Let Γ be a CD grammar system and let x, y be in V_i^*. The string x derives in G_i the string y using the *t-mode of derivation*, denoted $x \Rightarrow_{G_i}^{t} y$, iff $x \Rightarrow_{G_i}^{*} y$ and there is no $z, z \neq y$, with $y \Rightarrow_{G_i}^{*} z$.

Definition. If Γ is a CD grammar system then the *language generated* by Γ in the t-mode of derivation, denoted $L_t(\Gamma)$, is defined as the set of all words $z \in T^*$ for which there is a derivation

$$S = w_0 \Rightarrow_{G_{i_1}}^{t} w_1 \Rightarrow_{G_{i_2}}^{t} w_2 \Rightarrow_{G_{i_3}}^{t} \cdots \Rightarrow_{G_{i_r}}^{t} w_r = z.$$

Denote by CD_t the family of all languages generated by CD grammar systems in the t-mode of derivation.

Theorem 2. *$MPL_{REG} \subset CD_t$ and the inclusion is proper.*

Proof. First, assume that α is a pattern over $\Sigma \cup V$. Let $L_{x,\alpha}$ be the regular language corresponding to the variable x. For each $x \in V$, let $A_x = (\Sigma, Q_x, q_{0,x}, F_x, \delta_x)$ be a finite deterministic automaton such that $L(A_x) = L_{x,\alpha}$.

If $alph(\alpha) \cap V = \{x_1, \ldots, x_k\}$, $k \geqslant 0$, then consider the nonterminals: $[q,j]$, $q \in Q_{x_j}$, $1 \leqslant j \leqslant k$. Consider also, the morphism h defined by $h(a) = a, a \in \Sigma$ and $h(x_j) = [q_{0,x_j}, j]$, $1 \leqslant j \leqslant k$.

Then construct the CD grammar system Γ with the terminal alphabet Σ, the axiom S, and the nonterminal alphabet.

$$N = \{S\} \cup \{[q,j] \mid 1 \leqslant j \leqslant k,\ q \in Q_{x_j}\} \cup \{(q,j) \mid 1 \leqslant j \leqslant k,\ q \in Q_{x_j}\}$$

and the components having all the same alphabets N, T and the following sets of productions:

$$P_0 = \{S \to h(\alpha)\},$$

$$P_{j,a} = \{[q,j] \to a(\delta_{x_j}(q,a),j) \mid q \in Q_{x_j}\}, \quad 1 \leqslant j \leqslant k,\ a \in \Sigma,$$

$$P'_{j,a} = \{[q,j] \to a \mid q \in F_{x_j}\}, \quad 1 \leqslant j \leqslant k,\ a \in \Sigma,$$

$$P_j = \{(q,j) \to [q,j] \mid q \in Q_{x_j}\}, \quad 1 \leqslant j \leqslant k.$$

The component P_0 introduces a string $h(\alpha)$, obtained from α by replacing each variable x_j with $[q_{0,x_j}, j]$. The use of a component $P_{j,a}$ in the t-mode (when an enabled component works as long as possible) replaces all occurrences of $[q,j]$ by the same symbol $a \in \Sigma$. The components P_j return systematically the symbols (q,j) into $[q,j]$.

The derivation can be finished by components $P'_{j,a}$. The determinism of the automaton and the t-mode of the derivation ensure the fact that from each occurrence of a variable x_j in α we generate the same string. Consequently, $L_\Sigma(\alpha_1, \ldots, \alpha_n) = L_t(\Gamma)$.

Note that the family CD_t is closed under union (see [4]) and hence any language from MPL_{REG} is in CD_t.

The above inclusion is proper. This assertion follows from the fact that the language

$$L = \{a^n b^n \mid n \geqslant 1\}$$

is not in MPL_{REG}. Indeed, assume that $L = L_{\{a,b\}}(\alpha_1, \ldots, \alpha_n)$ for some patterns $\alpha_1, \ldots, \alpha_n$. For each $x \in V$ we have either $L_x \subseteq a^*$ or $L_x \subseteq b^*$ or $L_x \subseteq a^+ b^+$. If in one pattern α_i we have a variable of one of the first two types, then strings $a^n b^m$, $n \neq m$, can be produced. On the other hand, a variable x with $L_x \subseteq a^+ b^+$ cannot appear twice. Consequently the obtained strings are of the form $a^i a^n b^m b^j$ for $a^n b^m \in L_x$, for some $i, j \geqslant 0$. Such languages are regular, hence we cannot have $L = L_{\{a,b\}}(\alpha_1, \ldots, \alpha_n)$. □

Corollary. $MPL_{REG} \subset ETOL$.

Proof. It is known (see [4]) that $CD_t = ETOL$. □

Definition (*Ibarra* [7]). A *regular simple matrix grammar of degree n* is an ordered system $G = (N_1, \ldots, N_n, V, P, S)$, where N_i, $1 \leqslant i \leqslant n$, are finite sets of nonterminals, V is a terminal alphabet, S is the start symbol,

$$S \notin V \cup \bigcup_{i=1}^{n} N_i$$

and P is a finite set of n-dimensional vectors of rules, $(r_1, \ldots, r_n)$, such that each rule r_i is a regular rule over the alphabet $N_i \cup V$. Moreover, P contains also rules $(S \to u)$, with $u \in V^*$ and rules $(S \to u_0 X_1 u_1 \ldots X_n u_n)$, where $u_j \in V^*$, $j = 0, 1, \ldots, n$ and $X_i \in N_i$, $i = 1, \ldots, n$.

Let G be a regular simple matrix grammar of degree n. G defines a relation of *direct derivation* as

$$S \Rightarrow_G v \text{ iff } (S \to v) \in P$$

and $u_0 X_1 u_1 \ldots X_n u_n \Rightarrow_G u_0 v_1 u_1 \ldots v_n u_n$ iff $(X_1 \to v_1, \ldots, X_n \to v_n) \in P$, where $u_j \in V^*$, $j = 0, 1, \ldots, n$, $X_i \in N_i$, $i = 1, \ldots, n$, $v_k \in (V \cup N_k)^*$, $k = 1, \ldots, n$.

The *derivation relation* induced by G, denoted $\Rightarrow_G^*$, is the reflexive and transitive closure of $\Rightarrow_G$.

The *language generated* by a regular simple matrix grammar G of degree n is

$$L(G) = \{w \in V^* \mid S \Rightarrow_G^* w\}.$$

Notation. *RLSM* is the family of all regular simple matrix languages.

Theorem 3. $MPL_{REG} \subseteq RLSM$.

Proof. Assume that α is a pattern over $\Sigma \cup V$. For each $x \in V$, let $A_x = (\Sigma, Q_x, q_{0,x}, F_x, \delta_x)$ be a finite deterministic automaton such that $L(A_x) = L_{x,\alpha}$.

If $\alpha = \beta_0 x_1 \beta_1 \dots x_m \beta_m$, where $\beta_j \in \Sigma^*$, $x_i \in V$, $0 \leqslant j \leqslant m$, $1 \leqslant i \leqslant m$, then consider the nonterminals: $[q,j]$, $q \in Q_{x_j}$, $1 \leqslant j \leqslant m$. Denote

$$\alpha' = \beta_0 [q_{0,x_1}, 1] \beta_1 \dots [q_{0,x_m}, m] \beta_m.$$

Construct the right linear simple matrix grammar $G = (\Sigma, V_1, \dots, V_n, S, M)$ with $V_j = \{[q,j] \mid q \in Q_{x_j}\}$, $1 \leqslant j \leqslant m$, and M containing the following matrices:

(1) $(S \to \alpha')$.

(2) $([q_1, 1] \to \gamma_1 [\delta(q_1, \gamma_1), 1], \dots, [q_m, m] \to \gamma_m [\delta(q_m, \gamma_m), m])$, where $q_j \in Q_{x_j}$, $\gamma_j \in \{a, \lambda\}$, for some given $a \in \Sigma$, $1 \leqslant j \leqslant m$, such that $\gamma_{j_1} = \gamma_{j_2} = \dots = \gamma_{j_r} = a$ for $x_{j_1} = x_{j_2} = \dots = x_{j_r}$, $x_s \neq x_{j_1}$ for $s \notin \{j_1, \dots, j_r\}$ and $\gamma_s = \lambda$ for $s \notin \{j_1, \dots, j_r\}$.

(3) $([q_1, 1] \to \lambda, \dots, [q_m, m] \to \lambda)$, where $q_j \in F_{x_j}$, $1 \leqslant j \leqslant m$.

The determinism of the involved finite automata, the mode of derivation in right linear simple matrix grammars and the way of defining the matrices of G ensure the equality $L(G) = L_\Sigma(\alpha)$.

The family of right linear simple matrix languages is closed under union and hence any multi-pattern language is a right linear simple matrix language.

Moreover, the inclusion is proper. Consider again the language

$$L = \{a^n b^n \mid n \geqslant 1\},$$

which is a right linear simple matrix language but is not a multi-pattern language (see the second part of the proof of Theorem 3). □

Corollary. *Every language in MPL_{REG} is semilinear (hence the one-letter languages in MPL_{REG} are regular).*

Proof. The property holds for languages in $RLSM$. □

Corollary. *The emptiness and the finiteness of the intersection of a language in MPL_{REG} with a regular language is decidable. It is also decidable whether or not a language in MPL_{REG} is included in a regular language.*

Proof. The family $RLSM$ is closed under intersection with regular sets and the emptiness and finiteness problems are decidable for $RLSM$. As $L \subseteq R$ iff $L \cap (V^* - R) = \emptyset$, also the inclusion in a regular language is decidable. □

Remark. Consider now the particular case of the family MPL. From the preceding theorem we have

$$MPL \subset CD_t = ETOL,$$

$$MPL \subset RLSM.$$

The properness of these inclusions follows from Theorem 3, but a stronger assertion is true: there are regular languages not in MPL. For example $L = a^*b$ is not in MPL: the language is infinite, hence patterns with at least one variable are used, therefore $\{a,b\}^*$ must be included in $Sub(L)$ which is not true. (Here $Sub(L)$ denotes the set of subwords of the words in L.)

Some necessary conditions for a language L to be in MPL are:

(i) The language L has to be semilinear (consequence of Theorem 3).

(ii) If L is infinite, $L \subseteq \Sigma^*$, then $\Sigma^* \subseteq Sub(L)$.

(iii) If L is infinite, $L \subseteq \Sigma^*$, $card(\Sigma) \geqslant 2$, then L is not "slender", that is, there is no constant k such that, for all n, the set of words in L of length n is of cardinality $\leqslant k$.

4. Multi-pattern and context-free languages

In this section we investigate the interrelation between multi-pattern and context-free languages. Repetitions of the same variable induce a noncontext-free feature in pattern languages. On the other hand, very simple context-free (even regular) languages such as a^*b are not multi-pattern.

We will prove in this section that it is in general undecidable whether or not a given context-free language is multi-pattern. The proof, a reduction to the Post-Correspondence Problem, has some novel features which we believe are applicable also in similar situations elsewhere. In particular, our context-free languages associated to the given instance of the Post-Correspondence Problem are somewhat unusual.

Theorem 4. *It is not decidable whether or not an arbitrary given context-free language is in MPL.*

Proof. Take two arbitrary n-tuples of nonempty strings over the alphabet $\{a,b\}$, $\sigma = (\sigma_1, \ldots, \sigma_n)$, $\tau = (\tau_1, \ldots, \tau_n)$, and consider the following languages:

$$L_\gamma = \{b^{t_1}a^{i_1}b^{t_2}a^{i_2}\ldots b^{t_k}a^{i_k}c\gamma_{i_k}\ldots\gamma_{i_1} \mid k \geqslant 3,\ 1 \leqslant i_k \leqslant n,\ 1 \leqslant t_j \leqslant 3,\ t_j \equiv j \pmod 3,\ 1 \leqslant j \leqslant k\}$$

for $\gamma \in \{\sigma, \tau\}$,

$$L_S = \{w_1 c w_2 c z \tilde{w}_2 c \tilde{w}_1 \mid w_1, w_2, z \in \{a,b\}^*\},$$

$$L(\sigma,\tau) = \{a,b,c\}^* - ((L_\sigma\{c\}\{a,b\}^*\{c\}\tilde{L}_\tau) \cap L_S).$$

(Here $\tilde{w}_2$ denotes mirror image.) We just prove that $L(\sigma,\tau)$ is a context-free language and that it is equal to $\{a,b,c\}^*$ if and only if $PCP(\sigma,\tau)$ has no solution.

Assertion I. The language $L(\sigma,\tau)$ is a context-free language.

It is easy to observe that the language

$$((L_\sigma\{c\}\{a,b\}^*\{c\}\tilde{L}_\tau) \cap L_S)$$

is a deterministic context-free language and hence the complement of this language is a context-free language. Therefore, it follows Assertion I.

Now, clearly, when $L(\sigma,\tau)=\{a,b,c\}^*$, then $L(\sigma,\tau)$ is a multi-pattern language.

We shall prove that, if $L(\sigma,\tau)\neq\{a,b,c\}$, then $L(\sigma,\tau)$ is not a multi-pattern language, and this will end the proof.

Assume $L(\sigma,\tau)=L_{\{a,b,c\}}(\alpha_1,\ldots,\alpha_n)$, where $\alpha_1,\ldots,\alpha_n$ are patterns over $\{a,b,c\}\cup V$.

For every solution $(i_1,\ldots,i_k)$ of $PCP(\sigma,\tau)$, the strings:

$$ba^{i_1}b^2a^{i_2}b^3a^{i_3}ba^{i_4}b^2\ldots b^{t_k}a^{i_k}c\sigma_{i_k}\ldots\sigma_{i_1}c\delta c\tilde{\tau}_{i_1}\ldots\tilde{\tau}_{i_k}ca^{i_k}b^{t_k}\ldots a^{i_1}b \qquad (*)$$

are not in $L(\sigma,\tau)$. On the other hand, for all values of m, the language $L(\sigma,\tau)$ contains strings of the form

$$(b^{t_1}a^{i_1}\ldots b^{i_k}a^{i_k})^m\ldots(a^{i_k}b^{t_k}\ldots a^{i_1}b^{t_1})^m. \qquad (**)$$

In order to obtain the strings of the form (**) we need pattern $\alpha\in(\{a,b\}\cup V)^*$, $|\alpha|_V>0$.

Examine the possible form of these patterns.

(1) If there is a pattern $\gamma_1x\gamma_2$ with $\gamma_1,\gamma_2\in\{a,b\}^*$ then we must have

$$(b^{t_1}a^{i_1}\ldots b^{t_k}a^{i_k})^m=\gamma_1\gamma_1', \qquad (a^{i_k}b^{t_k}\ldots a^{i_1}b^{t_1})^m=\gamma_2'\gamma_2, \qquad (***)$$

for some words γ_1',γ_2' in $\{a,b\}^*$. For x replaced by

$$\gamma_1'c(\sigma_{i_k}\ldots\sigma_{i_1})^mcc(\tilde{\tau}_{i_1}\ldots\tilde{\tau}_{i_k})^mc\gamma_2'$$

we obtain a string of the form (*) (with $\delta=\lambda$), a contradiction.

Therefore, all patterns used in generating strings of the form (**) are of the form $\gamma_1x\gamma_3y\gamma_2$, with $\gamma_1,\gamma_2\in\{a,b\}^*$, $x,y\in V$, $\gamma_3\in(V\cup\{a,b\})^*$. Again γ_1,γ_2 must satisfy the condition (***).

(2) If there is such a pattern with $x\neq y$, then we can replace x with $\gamma_1'c(\sigma_{i_k}\ldots\sigma_{i_1})^mc$, y with $c(\tilde{\tau}_{i_1}\ldots\tilde{\tau}_{i_k})^mc\gamma_2'$ and irrespective of the form of γ_3, we obtain a string of the type (*) (with an arbitrary δ), a contradiction.

(3) In conclusion, all patterns used in generating strings of the form (**) are of the form $\gamma_1x\gamma_3x\gamma_2$, with $\gamma_1,\gamma_2,\gamma_3$ as above.

As γ_1,γ_2 are given (in a finite set of patterns) and m can be arbitrarily large, the string which replaces the two specified occurrences of x will contribute one to $(b^{t_1}a^{i_1}\ldots b^{t_k}a^{i_k})^m$ and the other to $(a^{i_k}b^{t_k}\ldots a^{i_1}b^{t_1})^m$. However, the substrings b^s appear on the left side in the order $b,b^2,b^3,b,\ldots$ and in the reverse order $b,b^3,b^2,b,\ldots$ on the right side. This implies that the two occurrences of x can introduce at most one substring b^s each, if we want to obtain a string of the form (**).

It follows that, in order to generate the strings (**), we have to essentially use the part γ_3 of the pattern, namely with γ_1x generating a prefix and $x\gamma_2$ a suffix of a string (**).

We continue now by examining the possible forms of γ_3.

If it is of the forms considered in cases (1), (2) above, then we obtain a contradiction in the same way.

If it is of the form in case (3) ($\gamma_3 = \gamma_{3,1} z \gamma_{3,3} z \gamma_{3,2}, \gamma_{3,1}, \gamma_{3,2} \in \{a,b\}^*$, $z \in V$, $\gamma_3 \in (\{a,b\} \cup V)^*$) then we continue the procedure. However, this can be done only finitely many times (the set of patterns is finite), hence eventually we either reach one of the cases (1), (2) – hence a contradiction – or we find a string over $\{a,b\}$, without variables. In this last case, only strings (**) with a bounded m can be produced – a contradiction which concludes the proof. □

Theorem 5. *It is not decidable whether or not an arbitrary given context-free language is in* MPL_{REG}.

Proof. Similar to the proof of Theorem 4. □

Open problems: Is it decidable whether or not: (1) a regular language is in MPL? (2) a language in MPL is a regular (context-free) language?

5. Hierarchies

We will now prove that a strictly increasing hierarchy of language families is obtained by increasing the number of generating patterns. This holds both in the E- and NE-case. Observe that, in spite of the overall equality

$$MPLE = MPLNE(= MPL),$$

there are differences between E- and NE-cases if only a fixed number of patterns is allowed. For instance, $(card(\Sigma))^2$ E-patterns are needed to generate the language generated by the single NE-pattern xy.

Theorem 6. *The number of patterns defines an infinite hierarchy*:

$$MPLE(n) \subset MPLE(n+1), \quad n \geqslant 1,$$

$$MPLNE(n) \subset MPLNE(n+1), \quad n \geqslant 1.$$

Proof. Consider the sequence of prime numbers

$$p_1 = 2,\ p_2 = 3,\ p_3 = 5,\ p_4 = 7, \ldots$$

the alphabet $\Sigma = \{a\}$ and the set of patterns

$$\pi_k = \{x^{p_1}, x^{p_2}, \ldots, x^{p_k}\}, \quad k \geqslant 1.$$

Clearly, a^{p_k+1} is in $L_{\Sigma,E}(\pi_{k+1})$ (replace x by a in x^{p_k+1}), but a^{p_k+1} is not in $L_{\Sigma,E}(\pi_k)$. The strings in $L_{\Sigma,E}(\pi_k)$ are either $\lambda, a^{p_1}, \ldots, a^{p_k}$ or their multiples (hence nonprime exponents). It is easy to see that $MPLE(k) \subset MPLE(k+1)$, $MPLNE(k) \subset MPLNE(k+1)$, $k \geqslant 1$, and that the inclusions are strict. □

6. Closure properties

We begin with the following simple observation.

Theorem 7. *For $L \subseteq \{a\}^*$, $L \in MPL$ if and only if L is regular.*

Proof. If $L \in MPL$, then L is regular, because the property holds for *RLSM* (see Section 3).

If $L \subseteq \{a\}^*$, L regular, then there is a finite set F and positive integers $p_1, \ldots, p_k$ and q such that

$$L = \{a^n \mid n \in F \text{ or } n = p_i + qj, j \geqslant 0, 1 \leqslant i \leqslant k\}.$$

Therefore, $L = L_{\{a\}}(\pi)$ for

$$\pi = \{a^n \mid n \in F\} \cup \{a^{p_i} x^q \mid 1 \leqslant i \leqslant k\}. \quad \square$$

The closure properties of MPL are summarized in the next theorem.

Theorem 8. *The family MPL is an anti-AFL, but it is closed under right/left derivatives. It is not closed under right/left quotients with regular sets, intersection and complement.*

Proof. The family MPL is not closed under any of the following operations:

Union: $a^+ \in MPL$, $\{b\} \in MPL$ but $a^+ \cup \{b\} \notin MPL$. (Note that if $L_1, L_2 \in MPL$ and $alph(L_1) = alph(L_2)$, then $L_1 \cup L_2 \in MPL$; construct the union of the patterns generating the two languages.)

Concatenation: $a^+ \in MPL$, $\{b\} \in MPL$ but $a^+\{b\} \notin MPL$. Again, if $alph(L_1) = alph(L_2)$, then $L_1 L_2 \in MPL$; rename the variable of the multi-pattern of L_2 and catenate each pattern corresponding to L_1 to each pattern corresponding to L_2.

Kleene +: $\{ab\} \in MPL$, but $\{ab\}^+ \notin MPL$.

Intersection with regular languages: $\Sigma^* \in MPL$ for all Σ; if $card(\Sigma) \geqslant 2$, there are regular languages over Σ which are not in MPL.

Morphisms: $a^+ \in MPL$, but $h(a^+) = \{ab\}^+ \notin MPL$, for $h(a) = ab$.

Inverse morphisms: $\{b\} \in MPL$, but $h^{-1}(b) = a^*ba^* \notin MPL$ for $h(a) = \lambda, h(b) = b$.

Left/right derivatives: Take $\alpha_1, \ldots, \alpha_n$ patterns over $\Sigma \cup V$ and $\beta \in \Sigma^+$. For every variable $x \in V$, consider the *replacement rules*

$$x \to \gamma, \quad \gamma \in \Sigma^*, \ 0 \leqslant |\gamma| < |\beta|,$$

$$x \to \gamma x, \quad \gamma \in \Sigma^*, \ |\gamma| = |\beta|.$$

For every pattern α_i consider all the patterns obtained by consistently applying an arbitrary set of such rules to α_i (every occurrence of some x is replaced by the same string γ or γx as above). We obtain in this way a set of patterns $\alpha'_1, \ldots, \alpha'_m$ such that $L_\Sigma(\alpha_1, \ldots, \alpha_n) = L_\Sigma(\alpha'_1, \ldots, \alpha'_m)$ and each α'_i is either in Σ^* or it has every variable

x with a left context in Σ^* of length at least $|\beta|$. Then clearly

$$\partial_\beta^l(L_\Sigma(\alpha_1, \ldots, \alpha_n)) = L_\Sigma(\partial_\beta^l(\alpha'_1, \ldots, \alpha'_m)).$$

which proves the closure under left derivatives. The case of the right derivatives is symmetric.

Left/right quotients by regular languages. Take $\alpha = xabax$, $\Sigma = \{a, b\}$ and the regular language $R = b^2a^+b^2$. Because

$$L_\Sigma(\alpha) = \{\beta aba\beta \mid \beta \in \{a, b\}^*\},$$

we have

$$R^{-1}L_\Sigma(\alpha) = \{\gamma abab^2a^nb^2\gamma \mid n \geqslant 1, \gamma \in \{a, b\}^*\}$$

(the prefix $b^2a^nb^2$ of a string $\beta aba\beta$ in $L_\Sigma(\alpha)$ must be a prefix of β, due to the presence of the string aba).

Assume that $R^{-1}L_\Sigma(\alpha) = L_\Sigma(\alpha_1, \ldots, \alpha_m)$ for some patterns $\alpha_1, \ldots, \alpha_m$ over $\{a, b\} \cup V$.

In every pattern we can replace every variable with aba, hence every pattern must contain at least two substrings b^2. At least one pattern must contain exactly two strings b^2, because

$$a^nabab^2a^pb^2a^n \in R^{-1}L_\Sigma(\alpha),$$

for all $n, p \geqslant 1$.

More exactly, there are patterns of the form $w_1abab^2w_2b^2w_3$, with w_1, w_2, $w_3 \in (V \cup \{a\})^*$. If all such patterns have $w_2 \in \{a\}^*$, then only finitely many strings of the form $a^nabab^2a^pb^2a^n$ can be obtained, for a given n. Consequently, there is a pattern $w_1abab^2w_2b^2w_3$ with w_2 containing a variable. Replace in this pattern all variables by aba. The obtained string is of the form $\beta_1abab^2\beta_2b\beta_3b^2\beta_4$ with $\beta_1, \beta_2, \beta_3, \beta_4 \in \{aba, c\}^*$, and such strings are not in $R^{-1}L_\Sigma(\alpha)$, a contradiction. The case of the right quotients is symmetric.

Intersection: Assume that: $\Sigma = \{a, b\}$, $\alpha_1 = xxab$, $\alpha_2 = xbax$. Then

$$L_\Sigma(\alpha_1) \cap L_\Sigma(\alpha_2) = \{\beta \in \Sigma^* \mid \text{there are } \gamma, \delta \in \Sigma^* \text{ such that } \beta = \gamma\gamma ab = \delta ba\delta\}.$$

Take the equation $\gamma\gamma ab = \delta ba\delta$. It follows that either $\delta = b$, $\gamma = b$ (hence $bbab \in L_\Sigma(\alpha_1) \cap L_\Sigma(\alpha_2)$) or $\delta = \delta_1\, ab$, hence $\gamma\gamma ab = \delta_1 abba\delta_1 ab$, $\gamma\gamma = \delta_1 abba\delta_1$. This implies $\gamma = \delta_1 ab$, $\gamma = ba\delta_1$, hence $\delta_1 ab = ba\delta_1$. Therefore, $\delta_1 = (ba)^kb$, for $k \geqslant 0$. Thus we obtain $\delta = (ba)^kbab = (ba)^{k+1}b$, $\gamma = (ba)^kbab = (ba)^{k+1}b$.

In conclusion,

$$L_\Sigma(\alpha_1) \cap L_\Sigma(\alpha_2) = \{(ba)^{k+1}b(ba)^{k+1}bab \mid k \geqslant 0\} \cup \{bbab\} = \{(ba)^kb(ba)^kbab \mid k \geqslant 0\}.$$

This language is not in MPL because, for instance, aa is not a subword of its strings.

Complement: The language $L = \{a, b\}^*ab\{a, b\}^*$ is in MPL but $\{a, b\}^* - L = b^*a^*$ does not contain the substring ab, hence it is not in MPL. □

Theorem 9. *The family MPL_{REG} is closed under union, concatenation morphisms, intersection with regular languages, but not closed under Kleene + and inverse morphisms.*

Proof. *Union*: Consider two multi-patterns

$$\pi_1 = \{\alpha_1, \ldots, \alpha_n\}, \qquad \pi_2 = \{\beta_1, \ldots, \beta_m\}$$

and take $\pi_2 = \{\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m\}$ with the same languages L_{x,α_i}, L_{x,β_j}. Then $L_\Sigma(\pi_3) = L_\Sigma({}_1\pi) \cup L_\Sigma(\pi_2)$.

Concatenation: Assume that $\pi_1 = \{\alpha_1, \ldots, \alpha_n\}$, $\pi_2 = \{\beta_1, \ldots, \beta_m\}$, replace all variables in π_2 with primed symbols and take $\pi_3 = \{\alpha_i\beta_j' \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m\}$, with the same languages L_{x,α_i} and with $L_{x',\beta_j} = L_{x,\beta_j}$. We have $L_\Sigma(\pi_3) = L_\Sigma(\pi_1)L_\Sigma(\pi_2)$.

Morphisms: Consider that $\pi = \{\alpha_1, \ldots, \alpha_n\}$ and $h: \Sigma^* \to \Sigma^*$. Take $h(\pi) = \{h(\alpha_1), \ldots, h(\alpha_n)\}$ and $L'_{x,\alpha_i} = h(L_{x,\alpha_i})$, $1 \leqslant i \leqslant n$. Clearly, $h(L_\Sigma(\pi)) = L_\Sigma(h(\pi))$.

Intersection with regular languages: Let $\pi = \{\alpha_1, \ldots, \alpha_n\}$ be a multi-pattern over $V \cup \Sigma$ with the regular languages associated to variables L_{x,α_i}, $1 \leqslant i \leqslant n$, $x \in V$. For $R \subseteq \Sigma^*$ a regular language, consider a deterministic finite automaton $A = (Q, \Sigma, q_0, F, \delta)$. For a pattern $\alpha_i = \beta_{i,1}x_{i,1} \ldots x_{i,k_i}\beta_{i,k_i+1}, \beta_{i,j} \in \Sigma^*$, $x_{i,j} \in V$ for all i,j, consider all the strings of the following form

$$\rho = \beta_{i,1}(q_1, x_{i,1}, q_1') \ldots (q_{k_i}, x_{i,k_i}, q_{k_i}')\beta_{i,k_i+1},$$

where

$$q_1 = \delta(q_0, \beta_{i,1}), \quad q_{j+1} = \delta(q_j', \beta_{i,j+1}), \qquad 1 \leqslant j \leqslant k_i - 1, \qquad \delta(q_{k_i}', \beta_{i,k_i+1}) \in F.$$

For each such string ρ and for each $x \in V$ appearing in it, let M be

$$M = \{(s, s') \mid (s, x, s') \text{ appears in } \rho\}$$

and define

$$L'_x = \bigcap_{(s,s') \in M} \{\gamma \in L_x \mid \delta(s, \gamma) = s'\}.$$

Note that these languages are all regular.

For each string ρ consider now the pattern obtained by replacing again all triples (s, x, s') with x. (For a given α_i we have more strings ρ, but for each ρ we obtain now only one pattern with languages L'_x associated as above.)

Denote by π' the set of all patterns obtained in this way. We have $L_\Sigma(\pi) \cap R = L_\Sigma(\pi')$.

Kleene +: The following language $L = \{ca^nca^nc \mid n \geqslant 1\}$ is in MPL because $L = L_\Sigma(cxcxc)$ with $L_x = a^+$. But, $L^+ \notin RLSM$, hence $L^+ \notin MPL_{REG}$.

Inverse morphisms: Assume that $\Sigma = \{a, b\}$, $\alpha = cxcx$, with $L_x = (ab)^+$, that is $L_\Sigma(\alpha) = \{c(ab)^nc(ab)^n \mid n \geqslant 1\}$. Consider also $h: \{a, b, c, d\} \to \{a, b, c\}^*$ defined by $h(a) = ca$, $h(b) = ba$, $h(c) = bc$, $h(d) = ab$. We obtain $h^{-1}(L_\Sigma(\alpha)) = \{ab^{n-1}cd^n \mid n \geqslant 1\}$, which is not in MPL_{REG} (the substrings b^{n-1}, d^n must be obtained using different variables, hence the powers cannot be related). □

7. Repetition-free multi-patterns

The study of multi-patterns is closely related to the study of *word rewriting systems with variables* (*WRSV*), see for instance, [10]. In particular, questions concerning the set $Red(R)$ of words *reducible* by a *WRSV*, R, can be expressed as questions concerning multi-pattern languages. For instance, the *ground reducibility problem* amounts to the problem of inclusion of a certain pattern language in a certain multi-pattern language. Usually, this leads to undecidable situations. However, many problems became decidable if the patterns involved are *repetition-free*, meaning that no variable appears twice in any given pattern. (The corresponding *WRSV*'s are often referred to as "linear".)

The following result is obvious.

Theorem 10. *Every repetition-free MPL is regular.*

We say that $(\alpha, \ldots, \alpha_n)$ is a *minimal* representation for an *MPL* $L = L(\alpha_1, \ldots, \alpha_n)$ if, for no i, $L(\alpha_i) \subseteq \bigcup_{j \neq i} L(\alpha_j)$. (Thus, every α_i is needed.) We do not know any instances of regular *MPL* languages not having a minimal repetition-free representation.

By Theorem 7, all decidability properties of regular languages concern also repetiton-free *MPL* languages. Particularly interesting from the point of view of pattern languages is the *decidability of finiteness of the complement*. It is also likely that every regular *MPL* language is effectively regular. (That is, if we know that an *MPL*, L, is regular, we can construct a regular expression for L.)

The following result is very interesting, in view of the undecidability of the inclusion for ordinary pattern languages.

Theorem 11. *The inclusion $K \subseteq L$ is decidable for pairs (K, L), where K and L are MPL and L is regular. Hence, it is decidable for MPL pairs (K, L), where L is repetition-free.*

Proof. (1) By the corollary of Theorem 3, K is *ETOL*. Hence, $K \subseteq L$ if and only if $K \cap \sim L = \emptyset$, where $\sim L$ means the complement of L. $K \cap \sim L$ is in *ETOL*, by the closure properties of *ETOL*. Hence, its emptiness is decidable.

The next proof provides an idea of a straight algorithm for the above problem.

(2) The proof uses the idea of a *finite test set*. We prove that we can compute from K and L a bound B such that if every work of K, obtained by assigning to each of the variables a word of length $\leqslant B$, is in L, then also $K \subseteq L$. Indeed, we claim that we can choose $B = q^p$, where q is the number of states in a deterministic finite automaton, *DFA*, accepting the complement of L, and p is the maximal number of occurrences of a single variable in one of the patterns defining K.

We proceed indirectly and assume that this test does not work. This means that $K \not\subseteq L$ but we do not find out this using words in the test set. In other words, there

exists some string $w \in L(\alpha) \cap \sim L$, where α is one of the patterns in K, but in order to get w, we have substituted a variable x in α by a word u with $k = |u| \geqslant q^p + 1$. We prove that we could as well use a shorter word u' for x and get a word $w' \in L(\alpha) \cap \sim L$.

The variable x occurs in w altogether $n \leqslant p$ times. We are interested in the corresponding n occurrences of the subword u in w:

$$w = \ldots u \ldots u \ldots u \ldots$$

When reading w from left to right, we observe after each letter the state DFA is in. When considering the n occurrences of u, we obtain in this way the n-tuples of states

$$(s_i^1, \ldots, s_i^n), \quad 1 \leqslant i \leqslant k.$$

Thus, the n-tuple $(s_k^1, \ldots, s_k^n)$ gives the state DFA is in after reading the last letter of each of the n occurrences of u.

Since $k > q^p \geqslant q^n$, there are i and $j, i < j$, such that $s_i^1 = s_j^1, \ldots, s_i^n = s_j^n$. This means that if the letters with numbers $i+1, \ldots, j$ (inclusive) are omitted from u, the resulting word u' satisfies the requirements states above. □

We consider repetition-free MPL's. The terminal words occurring in the patterns are finite in number but the patterns tell also the ordering of these terminal words. In some cases such an ordering is not necessary. We consider here the E-interpretation.

We say that an MPL language L has a *finite subword characterization* if L is defined by patterns of the form xwy, $w \in \Sigma^*$. For example, the MPL language $\Sigma^* a \Sigma^* b \Sigma^*$, $\Sigma = \{a, b\}$, has the finite subword characterization $\Sigma^* ab \Sigma^*$. The MPL language $\Sigma^* a \Sigma^* b \Sigma^* a \Sigma^*$ has no finite subword characterization.

Theorem 12. *It is decidable whether or not a given regular (hence, a given repetition-free) MPL language L has a finite subword characterization.*

Proof. If $w_1, \ldots, w_k$ are the words used in the finite subword characterization, we may assume that none of them is a proper subword of the other. This follows because if w_j is a proper subword of w_i, then

$$L_E(xw_iy) \subset L_E(xw_jy)$$

and thus w_i can be omitted. Thus, we have to find out whether L contains infinitely many words with this property. This is the case iff

$$L \cap \Sigma(\sim L) \cap (\sim L)\Sigma$$

is infinite. For a regular L, this is a decidable property. □

The preceding theorem appears in [10] in a formulation dealing with linear $WRSV$'s.

8. Conclusion. Ambiguity

Numerous other aspects of multi-pattern languages remain to be investigated. Of particular interest and importance are issues concerning *ambiguity*. We hope to return to this topic in a forthcoming contribution.

Ambiguity can be defined in the natural way both for patterns and multi-patterns, as well as for the generated languages. Thus, an NE-pattern α is *unambiguous* iff, for every word $w \in L_{NE}(\alpha)$, there is a unique substitution for the variables in α giving rise to w. A pattern (resp. multi-pattern) language is *unambiguous* iff it can be generated by an unambiguous pattern (resp. multi-pattern). *Degrees* of *ambiguity* can be introduced in the usual way.

An NE-pattern α is unambiguous iff the language $L = L_{NE}(\alpha)$ is unambiguous. This follows because every NE-pattern β satisfying $L = L_{NE}(\beta)$ results from α by a renaming of the variables. An analogous statement does not hold for E-patterns. For instance, the E-pattern xy is ambiguous (of degree infinity), whereas the language $L_E(xy) = L_E(x)$ is unambiguous.

It is easy to see that every pattern containing occurrences of a single variable is unambiguous. On the other hand, a pattern is ambiguous if it contains occurrences of at least two variables but at most one terminal, or occurrences of at least two variables, one of which occurs only once in the pattern. We conjecture that all problems dealing with the ambiguity of multi-patterns and their languages are decidable.

References

[1] J. Albert and L. Wegner, Languages with homomorphic replacements, in: *ICALP-80 Proc.*, Lecture Notes in Computer Science, Vol. 85 (Springer, Berlin, 1980) 19–29.
[2] D. Angluin, Finding patterns common to a set of strings, *J Comput. System Sci.* **21** (1980) 46–62.
[3] J. Bean, A. Ehrenfeucht and G. McNulty, Avoidable patterns in strings of symbols, *Pacific J. Math.* **85** (1979) 261–294.
[4] E. Csuhaj-Varju, J. Dassow, J. Kelemen and Gh. Păun, *Grammar Systems* (Gordon and Breach, London, 1994).
[5] J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory* (Springer, Berlin, 1989).
[6] J. Dassow, Gh. Păun and A. Salomaa, Grammars based on patterns, *Internat. J. Found. Comput. Sci.* **4** (1993) 1–14.
[7] O. Ibarra, Simple matrix grammars, *Inform. and Control* **17** (1970) 359–394.
[8] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa and S. Yu, Pattern languages with and without erasing, *Internat. J. Comput. Math.*, to appear.
[9] T. Jiang, A. Salomaa, K. Salomaa and S. Yu, Inclusion is undecidable for pattern languages in: *ICALP-93 Proc.*, Lecture Notes in Computer Science (Springer, Berlin) to appear.
[10] G. Kucherov and M. Rusinowitch, On ground reducidibility problem for word rewriting systems with variables, Report CRIN 93-R-012, Centre de Recherhe en Informatique de Nancy.
[11] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems* (Academic Press, New York, 1980).
[12] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).
[13] A. Thue, Über unendliche Zeichenreihen, Norske Vid. Selsk. Skr., I Mat. Nat. Kl. Kristiania 7 (1906) 1–22.